

Javascript Assignment

Promises :-

Promises in JS are just like promises in real life .

For Example in real life you may go to tailor and asked him to sew suit he promise you to give that suit to you after one day , Obviously You are not going to wait till he give you the suit you will go and come after day after time completion . Same is the case here that you request some function which require some time you are not going wait ignoring whole remaining code you will come to the that code after some time when and take the output .

Here is the Syntax of Promise :-

```
let p = new Promise (function(reject , resolve)=>{  
  //code  
})
```

There are always two possibilities of a promises

1. Resolve
2. Reject

There are two states of a promise :-

3. Pending
4. Fullfilled

async and await :-

By using await we can temporarily stop execution in a function but we can only use await in a async function . We can understand better by example .

```
let p = new Promise (function(reject , resolve)=>{
  setTimeout(()=>{
    return 56
  },2000)
})
const newFun = async () =>{
  let a = await p
  console.log(a)
  console.log("helo World")
}
```

In above Example hello world will print after the value of a but if we don't use await then hello world will print first then value of a .

Event Loop :-

It is the flow of control in js we can understand easily by a diagram .

When we run a command in js it goes in the task queue from that queue it has two option whether into call stack or into web api for example if a function which needs time like `setTimeout` it will go to the web Api and after completion of time it will go to call stack .

